

caBIG Identifiers: Personal Position Paper
George Komatsoulis
v. 2.0 Draft
02-Dec-04

- I. Usage of Identifiers – Schemas:** Schemas should be marked with an identifier, and any change in the schema should trigger a change in the identifier. Schema identifiers should be versioned in my opinion; if it is decided that they should not be versioned than any changes to schemas would have to trigger a change to the schema ID. If versioned, it is essential to develop guidance with regard to determining the sort of changes that trigger a ‘versioning’ change as opposed to a ‘new id’ change.
- II. Usage of Identifiers – Data Providers:** I concur with the suggestion that an identifier be attached to any persistent molecule of data. I define a persistent molecule as the information contained in one instance of a data object, derived from a data store managed by a grid node, whose UML model is registered in the caDSR. Thus, identifiers would not be attached to rows in underlying tables, per se, only in the assembled data objects served to grid users. Note that this definition still requires unique identifiers on data molecules that are generated dynamically from multiple tables. In that case it is the responsibility of the data content provider to identify an appropriate primary key so that a future request returns identical information for unchanged data, or points to the same ‘data aggregate’ if the data is evolving. It might also be useful to attach some sort of tag (whether an identifier or other form of marking) to the item defined by the data source as the ‘primary key’.

If the Grid service is returning complex XML (i.e. if the molecule of data itself contains complete data molecules registered in the caDSR), the provider should return the identifier attached to the primary molecule, as well as the identifier(s) that are attached to all of the secondary (or tertiary, etc.) data molecules as well. In that case, the identifier should be attached to the subsidiary data molecules in the same fashion as the primary is attached to the primary data molecule.

It is my opinion that it is beyond the scope of our job to provide a data object identifier that indicates that two data objects from different sources represent properties of the same physical entity. In other words, I think that it is unnecessary to demand that the data object identifier communicate that two records refer to the same gene, patient, etc. A separate identifier (but perhaps following our standard) should be used for such identification needs. In those cases, cooperating workspaces, cancer centers, etc. should be given a specific source authority and control of the issuance of those identifiers. See below for an example of such an identifier.

III. Usage of Identifiers – Analytical Service Providers: Aggregated analytical service results based on data supplied by users should not have persistent identifiers attached to the primary record, even though the structure of that record is a caDSR registered data object. The logic behind this argument is that the analytical service provider has no means (short of overly onerous logging requirements) to identify identical output sets. However, if the results that are aggregated into the primary record are themselves persistent data molecules (as above), they should be identified appropriately. An example of these proposed rules would be a BLAST service. The primary record (a BLAST result) would not be required to have a unique ID, but the individual sequence records sent back with the BLAST result are required to have the unique identifiers (Genbank accessions/gid's) in this case.

IV. Characteristics of Identifiers: Identifiers should be globally unique and at least partially resolvable by any interested party on the grid. By partially resolvable, I mean that the source authority (i.e. the group that maintains control of the data) should be identifiable so that future requests could be properly directed to the source. In addition, the identifiers should probably contain a resolvable component indicating the ID of the schema. This would allow a system to rapidly determine that two results are from different sources but are (by definition) symantically equivalent. On consideration, I suspect that versioning should be supported, but not required. Thus, data sources that desire to maintain older versions can, while those that want to keep only the current data can do so. An alternative would be use a data creation/modification timestamp as the version. This would allow for new data checking without requiring a formal version number. In all cases, an unversioned identifier would refer to the most recent value of the datum. Thus, an ID for a schema might look like:

```
source_authority_id.schema_id.schema_vers
```

For a schema, the source authority ID should probably be caBIG as a whole. An ID for an instance of data might look like:

```
source_authority.schema_id.schema_vers.node_defined_id.[node_vers]
```

An ID to uniquely identify a gene, patient, etc. (i.e. an id for a specific datum) might look like this:

```
source_authority.datatype_id.datatype_vers.node_defined_id.[node_vers]
```